IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of

      Nadeem SYED, et al.   :

                          :

Serial No.: NYA   :   Group Art Unit: NYA

                          :

Filed: December 13, 2001   :   Examiner: NYA

                          :

For: METHOD AND APPARATUS FOR :
       COPYING DATA THAT RESIDES IN :
       A DATABASE   :

PRELIMINARY AMENDMENT

FOR REISSUE PATENT APPLICATION

Hon. Commissioner of Patents
BOX REISSUE
Washington, D.C. 20231

Sir:

Prior to examination of the above-referenced reissue patent application, please amend the application as follows:

IN THE CLAIMS

Please **amend** Claims 1-6, 8-11, 13, 14, 16-25, 27, 30-35, 37, 40, and 41 as shown below.

Please **add** Claims 42-89 as shown below.

No Claims have been cancelled.

Therefore, Claims 1-89 are **pending** in the application.

1   1.   (Amended) A method for supplying a consistent set of data to a software application, the

2   method comprising the steps of:

3   [launching said software application;]

4   identifying a particular set of data that is required by the software application;

5   [requesting a first] causing each process of a first set of two or more processes to obtain a

6   snapshot time from a database server associated with a first database, wherein the

7   snapshot times obtained by the first set of processes are associated with a same

8   state of the particular set of data [time causes all subsequent reads of said first

9   database by the first process to return data that reflects a database state associated

10   with the snapshot time];

11   after [the first] each process of the first set of processes obtains [the] a snapshot time,

12   causing the first [process] set of processes to extract the particular set of data from

13   the first database; [and]

14   [supplying said software application with the particular set of data that was extracted

15   from the first database.]

16   causing the first set of processes to store a first copy of the particular set of data that is

17   extracted from the first database in one or more files that are separate from the

18   first database;

19   causing a second set of one or more processes to store a second copy of the particular set

20   of data that is stored in the one or more files, wherein the second copy is stored

21   separate from the first database, and wherein no process in the second set of

22   processes is in the first set of processes; and

23   allowing the software application to access the second copy of the particular set of data.

1   2.   (Amended) The method of claim 1, [further comprising the step of causing a second

2   process to store the particular set of data] wherein the second copy is stored in a second

3   database.

1   3.   (Amended) The method of claim 2, wherein [the step of causing the second process to

2   store the particular set of data in the second database includes the steps of]:

3   [writing the particular set of data to] the one or more files include one or more flat files;

4   and

5   [executing a loader process,] <u>the second set of processes is a set of one or more loader</u>

6   <u>processes</u>, wherein the <u>set of</u> loader [process] <u>processes</u> loads the <u>second copy of</u>

7   <u>the</u> particular set of data <u>into the second database based on the first copy that is</u>

8   <u>stored in</u> [from] the one or more flat files [to the second database].

1   4.   (Amended) The method of claim 3, [wherein the step of writing the particular set of data

2   to one or more flat files includes] <u>further comprising</u> the steps of:

3   <u>each process of</u> the first <u>set of</u> [process] <u>processes</u> informing a coordinator process when

4   [it] <u>said each process</u> has finished writing data to a particular flat file <u>of the one or</u>

5   <u>more flat files</u>; and

6   <u>after each process of the first set of processes informs the coordinator process,</u> the

7   coordinator <u>process</u> [using the information to tell the] <u>informing each</u> loader

8   process <u>of the set of loader processes</u> when [it] <u>said each loader process</u> can begin

9   loading <u>the second copy of the particular set of data into the second database</u>

10   <u>based on each of</u> the <u>one or more</u> flat [file] <u>files</u> [into the second database].

1   5.   (Amended) The method of claim 3, wherein [the step of]:

2   [writing the particular set of data to the flat file includes the step of writing the particular

3   set of data to] <u>the one or more flat files comprises</u> a plurality of flat files; and

4   [executing the loader process includes the step of executing] <u>the set of loader processes</u>

5   <u>comprises</u> a plurality of loader processes, wherein the plurality of loader

6   processes [load] <u>loads</u> the particular set of data from the plurality of flat files to

7   the second database.

1   6.   (Amended) The method of claim 2, [wherein the step of supplying said software

2   application with data from said particular set of data includes] <u>further comprising</u> the

3   steps of:

4   said software application reading the <u>second copy of the</u> particular set of data stored in

5   the second database; and

6   said software application generating a planning schedule based on the <u>second copy of the</u>

7   particular set of data.

1   7.   (Not Amended) The method of claim 1, wherein the step of identifying the particular set
2   of data includes the step of creating a copy table list, wherein the copy table list contains
3   entries that identify the particular set of data in the first database.

1   8.   (Amended) The method of claim 7, further comprising the steps of:
2   executing a <u>set of one or more</u> delete [process] <u>processes</u>, wherein the <u>set of</u> delete
3   [process] <u>processes</u> uses the copy table list to identify data that needs to be deleted
4   in a second database; and
5   deleting the identified data from the second database.

1   9.   (Amended) The method of claim 7, [where] <u>wherein</u> the step of creating the copy table
2   list includes the steps of:
3   communicating with the software application to [identifying] <u>identify</u> a set of planning
4   data, [where] <u>wherein</u> the planning data is required for generating a planning
5   schedule; and
6   creating the copy table list based on the identified set of planning data.

1   10. (Amended) The method of claim 1, wherein [the step of supplying said software
2   application with data from said particular set of data includes the steps of:
3   writing the particular set of data to] <u>the one or more files include</u> one or more flat files[;],
4   and <u>the method further comprises the steps of:</u>
5   [supplying] <u>allowing the software application to access</u> the one or more flat files [to said
6   software application, wherein] <u>such that</u> said software application [generates] <u>can</u>
7   <u>generate</u> a planning schedule based on information contained in the one or more
8   flat files.

1   11. (Amended) A method for producing a copy of data from a first database, the method
2   comprising the steps of:
3   locking a first set of data in the first database;
4   after locking the first set of data,

5   requesting a plurality of processes to obtain snapshot times from a database server

6   associated with said first database, wherein the snapshot times cause all

7   subsequent reads of the first database by the plurality of processes to

8   return data from the first database as of said snapshot times;

9   waiting a particular period of time for the plurality of processes to be assigned

10   snapshot times;

11   releasing the locks on the first set of data in the first database;

12   using a successful set of said plurality of processes to extract a copy of the first set

13   of data from the first database, wherein said successful set includes only

14   those processes of the plurality of processes that were assigned a snapshot

15   time within the particular period of time; and

16   storing the copy of the first set of data separate from said first <u>set</u> of data.

1   12.   (Not Amended) The method of claim 11, wherein the step of identifying the first set of

2   data includes the step of creating a copy table list, wherein the copy table list contains

3   entries that identify the first set of data in the first database.

1   13.   (Amended) The method of claim 12, [where] <u>wherein</u> the step of creating the copy table

2   list includes the steps of:

3   identifying a set of planning data, [where] <u>wherein</u> the planning data is required to

4   generate a planning schedule; and

5   creating the copy table list based on the planning data required to generate the planning

6   schedule.

1   14.   (Amended) The method of claim 12, further comprising the steps of:

2   executing a plurality of delete processes, wherein the plurality of delete processes [use]

3   <u>uses</u> the copy table list to identify data that needs to be deleted in a second

4   database; and

5   deleting the identified data from the second database.

1   15.   (Not Amended) The method of claim 11, wherein the step of storing the copy of the first

2   set of data includes the steps of:

3   writing the copy of the first set of data to a plurality of flat files; and

4   executing a plurality of loader processes, wherein the plurality of loader processes load

5   the copy of the first set of data from the plurality of flat files to a second database.

1   16.   (Amended) The method of claim 15, wherein:

2   the steps of writing the copy of the first set of data to [a] the plurality of flat files further

3   includes the step of notifying a coordinator process that data has been written to

4   one of the plurality of flat files; and

5   the steps of executing the plurality of loader processes further includes the step of the

6   coordinator, upon being notified that data has been written to one of the plurality

7   of flat files, launching a loader process to load the first set of data from one of the

8   plurality of flat files to the second database.

1   17.   (Amended) The method of claim 15, wherein the step of writing the copy of the first set

2   of data to the plurality of flat files includes the steps of:

3   the plurality of [process] processes informing a coordinator process when [it] the plurality

4   of processes has finished writing data to a particular flat file; and

5   the coordinator process using the information to tell one of the plurality of loader

6   processes when [it] said one of the plurality of loader processes can begin loading

7   the particular flat file into the second database.

1   18.   (Amended) The method of claim 11, wherein the step of requesting the plurality of

2   processes to obtain [a] snapshot [time] times includes the step of requesting the plurality

3   of processes based on a user input parameter, wherein the user input parameter identifies

4   how many processes should be requested to obtain [a] snapshot [time] times.

1   19.   (Amended) The method of claim 11, wherein the plurality of processes is a plurality of

2   snapshot processes, and wherein the step of extracting the copy of the first set of data

3   from the database includes the steps of:

4   assigning a set of copy data to the plurality of snapshot processes; and

5   retrieving data from the first database based on the set of copy data that was assigned to

6   the plurality of snapshot processes.

1   20.   (Amended) The method of claim 11, wherein the step of storing the copy of the first set
2   of data includes the steps of storing the copy of the first set of data as blob files that are
3   separate from said first set of data.

1   21.   (Amended) The method of claim 11, wherein the step of storing the copy of the first set
2   of data includes the steps of storing the copy of the first set of data in said first [of data]
3   database.

1   22.   (Amended) A computer-readable medium carrying one or more sequences of one or
2   more instructions for supplying a consistent set of data to a software application, the one
3   or more sequences of one or more instructions including instructions which, when
4   executed by one or more processors, cause the one or more processors to perform the
5   steps of:
6   [launching said software application;]
7   identifying a particular set of data that is required by the software application;
8   [requesting a first] causing each process of a first set of two or more processes to obtain a
9   snapshot time from a database server associated with a first database, wherein the
10   snapshot times obtained by the first set of processes are associated with a same
11   state of the particular set of data [time causes all subsequent reads of said first
12   database by the first process to return data that reflects a database state associated
13   with the snapshot time];
14   after [the first] each process of the first set of processes obtains [the] a snapshot time,
15   causing the first [process] set of processes to extract the particular set of data from
16   the first database; [and]
17   [supplying said software application with the particular set of data that was extracted
18   from the first database.]
19   causing the first set of processes to store a first copy of the particular set of data that is
20   extracted from the first database in one or more files that are separate from the
21   first database;

22   causing a second set of one or more processes to store a second copy of the particular set

23   of data that is stored in the one or more files, wherein the second copy is stored

24   separate from the first database, and wherein no process in the second set of

25   processes is in the first set of processes; and

26   allowing the software application to access the second copy of the particular set of data.

1   23.   (Amended) The computer-readable medium of claim 22, [wherein the computer-

2   readable medium further comprises instructions for performing the step of causing a

3   second process to store the particular set of data] wherein the second copy is stored in a

4   second database.

1   24.   (Amended) The computer-readable medium of claim 23, wherein [the step of causing the

2   second process to store the particular set of data in the second database includes the steps

3   of]:

4   [writing the particular set of data to] the one or more files include one or more flat files;

5   and

6   [executing a loader process,] the second set of processes is a set of one or more loader

7   processes, wherein the set of loader [process] processes loads the second copy of

8   the particular set of data into the second database based on the first copy that is

9   stored in [from] the one or more flat files [to the second database].

1   25.   (Amended) The computer-readable medium of claim 24, [wherein the step of writing the

2   particular set of data to one or more flat files includes] further comprising instructions for

3   performing the steps of:

4   each process of the first set of [process] processes informing a coordinator process when

5   [it] said each process has finished writing data to a particular flat file of the one or

6   more flat files; and

7   after each process of the first set of processes informs the coordinator process, the

8   coordinator process [using the information to tell the] informing each loader

9   process of the set of loader processes when [it] said each loader process can begin

10   loading the second copy of the particular set of data into the second database

11   based on each of the one or more flat [file] files [into the second database].

1 26. (Not Amended) The computer-readable medium of claim 22, wherein the step of

2   identifying the particular set of data includes the step of creating a copy table list,

3   wherein the copy table list contains entries that identify the particular set of data in the

4   first database.

1 27. (Amended) A computer-readable medium carrying one or more sequences of one or

2   more instructions for producing a copy of data from a first database, the one or more

3   sequences of one or more instructions including instructions which, when executed by

4   one or more processors, cause the one or more processors to perform the steps of:

5   locking a first set of data in the first database;

6   after locking the first set of data,

7   requesting a plurality of processes to obtain snapshot times from a database server

8   associated with said first database, wherein the snapshot times cause all

9   subsequent reads of the first database by the plurality of processes to

10   return data from the first database as of said snapshot times;

11   waiting a particular period of time for the plurality of processes to be assigned

12   snapshot times;

13   releasing the locks on the first set of data in the first database;

14   using a successful set of said plurality of processes to extract a copy of the first set

15   of data from the first database, wherein said successful set includes only

16   those processes of the plurality of processes that were assigned a snapshot

17   time within the particular period of time; and

18   storing the copy of the first set of data separate from said first set of data.

1 28. (Not Amended) The computer-readable medium of claim 27, wherein the step of

2   identifying the first set of data includes the step of creating a copy table list, wherein the

3   copy table list contains entries that identify the first set of data in the first database.

1 29. (Not Amended) The computer-readable medium of claim 27, wherein the step of storing

2   the copy of the first set of data includes the steps of:

3   writing the copy of the first set of data to a plurality of flat files; and

4   executing a plurality of loader processes, wherein the plurality of loader processes load

5   the copy of the first set of data from the plurality of flat files to a second database.

1   30.   (Amended) The computer-readable medium of claim 29, wherein the step of writing the

2   copy of the first set of data to the plurality of flat files includes the steps of:

3   the plurality of [process] <u>processes</u> informing a coordinator process when [it] <u>the plurality</u>

4   <u>of processes</u> has finished writing data to a particular flat file; and

5   the coordinator <u>process</u> using the information to tell one of the plurality of loader

6   processes when [it] <u>said one of the plurality of loader processes</u> can begin loading

7   the particular flat file into the second database.

1   31.   (Amended) The computer-readable medium of claim 27, <u>wherein the plurality of</u>

2   <u>processes is a plurality of snapshot processes, and</u> wherein the step of extracting the copy

3   of the first set of data from the database includes the steps of:

4   assigning a set of copy data to the plurality of snapshot processes; and

5   retrieving data from the first database based on the set of copy data that was assigned to

6   the plurality of snapshot processes.

1   32.   (Amended) A computer system for supplying a consistent set of data to a software

2   application, the computer system comprising:

3   a memory;

4   one or more processors coupled to the memory; and

5   a set of computer instructions contained in the memory, the set of computer instructions

6   including computer instructions which when executed by the one or more

7   processors, cause the one or more processors to perform the steps of:

8   [launching said software application;]

9   identifying a particular set of data that is required by the software application;

10   [requesting a first] <u>causing each</u> process <u>of a first set of two or more processes</u> to

11   obtain a snapshot time from a database server associated with a first

12   database, wherein the snapshot <u>times obtained by the first set of processes</u>

13   <u>are associated with a same state of the particular set of data</u> [time causes

14   all subsequent reads of said first database by the first process to return data

15   that reflects a database state associated with the snapshot time];

16   after [the first] each process of the first set of processes obtains [the] a snapshot

17   time, causing the first [process] set of processes to extract the particular

18   set of data from the first database; [and]

19   [supplying said software application with the particular set of data that was

20   extracted from the first database.]

21   causing the first set of processes to store a first copy of the particular set of data

22   that is extracted from the first database in one or more files that are

23   separate from the first database;

24   causing a second set of one or more processes to store a second copy of the

25   particular set of data that is stored in the one or more files, wherein the

26   second copy is stored separate from the first database, and wherein no

27   process in the second set of processes is in the first set of processes; and

28   allowing the software application to access the second copy of the particular set of

29   data.

1   33.   (Amended) The computer system of claim 32, [further including instructions for

2   performing the step of causing a second process to store the particular set of data]

3   wherein the second copy is stored in a second database.

1   34.   (Amended) The computer system of claim 33, wherein [the step of causing the second

2   process to store the particular set of data in the second database includes the steps of]:

3   [writing the particular set of data to] the one or more files include one or more flat files;

4   and

5   [executing a loader process,] the second set of processes is a set of one or more loader

6   processes, wherein the set of loader [process] processes loads the second copy of

7   the particular set of data into the second database based on the first copy that is

8   stored in [from] the one or more flat files [to the second database].

1   35.   (Amended) The computer system of claim 34, [wherein the step of writing the particular

2   set of data to one or more flat files includes] <u>further comprising instructions for</u>

3   <u>performing</u> the steps of:

4   <u>each process of</u> the first <u>set of</u> [process] <u>processes</u> informing a coordinator process when

5   [it] <u>said each process</u> has finished writing data to a particular flat file <u>of the one or</u>

6   <u>more flat files</u>; and

7   <u>after each process of the first set of processes informs the coordinator process,</u> the

8   coordinator <u>process</u> [using the information to tell the] <u>informing each</u> loader

9   process <u>of the set of loader processes</u> when [it] <u>said each loader process</u> can begin

10   loading <u>the second copy of the particular set of data into the second database</u>

11   <u>based on each of</u> the <u>one or more</u> flat [file] <u>files</u> [into the second database].

1   36.   (Not Amended) The computer system of claim 32, wherein the step of identifying the

2   particular set of data includes the step of creating a copy table list, wherein the copy

3   table list contains entries that identify the particular set of data in the first database.

1   37.   (Amended) A computer system for producing a copy of data from a first database, the

2   computer system comprising:

3   a memory;

4   one or more processors coupled to the memory; and

5   a set of computer instructions contained in the memory, the set of computer instructions

6   including computer instructions which when executed by the one or more

7   processors, cause the one or more processors to perform the steps of:

8   locking a first set of data in the first database;

9   after locking the first set of data,

10   requesting a plurality of processes to obtain snapshot times from a

11   database server associated with said first database, wherein the

12   snapshot times cause all subsequent reads of the first database by the

13   plurality of processes to return data from the first database as of said

14   snapshot times;

15   waiting a particular period of time for the plurality of processes to be

16   assigned snapshot times;

17   releasing the locks on the first set of data in the first database;

18   using a successful set of said plurality of processes to extract a copy of the

19   first set of data from the first database, wherein said successful set

20   includes only those processes of the plurality of processes that were

21   assigned a snapshot time within the particular period of time; and

22   storing the copy of the first set of data separate from said first <u>set</u> of data.

1   38.   (Not Amended) The computer system of claim 37, wherein the step of identifying the

2   first set of data includes the step of creating a copy table list, wherein the copy table list

3   contains entries that identify the first set of data in the first database.

1   39.   (Not Amended) The computer system of claim 37, wherein the step of storing the copy

2   of the first set of data includes the steps of:

3   writing the copy of the first set of data to a plurality of flat files; and

4   executing a plurality of loader processes, wherein the plurality of loader processes load

5   the copy of the first set of data from the plurality of flat files to a second database.

1   40.   (Amended) The computer system of claim 39, wherein the step of writing the copy of the

2   first set of data to the plurality of flat files includes the steps of:

3   the plurality of [process] <u>processes</u> informing a coordinator process when [it] <u>the plurality</u>

4   <u>of processes</u> has finished writing data to a particular flat file; and

5   the coordinator <u>process</u> using the information to tell one of the plurality of loader

6   processes when [it] <u>said one of the plurality of loader processes</u> can begin loading

7   the particular flat file into the second database.

1   41.   (Amended) The computer system of claim 37, <u>wherein the plurality of processes is a</u>

2   <u>plurality of snapshot processes, and</u> wherein the step of extracting the copy of the first set

3   of data from the database includes the steps of:

4   assigning a set of copy data to the plurality of snapshot processes; and

5   retrieving data from the first database based on the set of copy data that was assigned to

6   the plurality of snapshot processes.

1   42.   (New) A method for extracting data from a first database, the method comprising the

2   steps of:

3   causing a set of two or more processes to obtain snapshot times that cause all subsequent

4   reads of the first database by each process in the set of processes to return data

5   from a same state of at least a portion of the first database as will be returned to

6   all other processes in the set of processes; and

7   causing the set of processes to extract a copy of the portion of the first database.

1   43.   (New) The method of claim 42, wherein the set of processes are included in a plurality of

2   processes, and the step of causing the set of processes to obtain snapshot times comprises

3   the step of:

4   causing the plurality of processes to attempt to obtain snapshot times that cause all

5   subsequent reads of the first database by each process of the plurality of processes

6   that successfully obtains one of the snapshot times to return data from the same

7   state of at least the portion of the first database as will be returned to all other

8   processes of the plurality of processes that successfully obtain one of the snapshot

9   times.

1   44.   (New) The method of claim 43, wherein at least one process of the plurality of processes

2   fails to one of the snapshot times within a particular period of time.

1   45.   (New) The method of claim 42, wherein each process of the set of processes obtains a

2   snapshot time by sending a request for a snapshot time to a database management system

3   that is associated with the first database and that assigns a snapshot time in response to

4   the request.

1   46.   (New) The method of claim 42, wherein each process of the set of processes is assigned

2   to extract a particular subset of the portion from the first database.

1   47.   (New) The method of claim 42, wherein each process of the set of processes obtains a

2   snapshot time within a particular period of time.

1   48.   (New) The method of claim 47, wherein the set of processes does not begin to extract the

2   copy of the portion until after expiration of the particular period of time.

1   49.   (New) The method of claim 47, wherein at least one process of the set of processes

2   begins to extract the copy of the portion prior to expiration of the particular period of

3   time.

1   50.   (New) The method of claim 47, further comprising the step of:

2   locking the portion of the first database during the particular period of time.

1   51.   (New) The method of claim 42, further comprising the step of:

2   using a set transaction read only command such that each process of the set of processes

3   can obtain a snapshot time.

1   52.   (New) The method of claim 42, further comprising the step of:

2   storing the copy separate from the portion.

1   53.   (New) The method of claim 52, wherein the step of storing the copy comprises the step

2   of:

3   storing the copy in a second database.

1   54.   (New) The method of claim 53, wherein step of storing the copy in the second database

2   comprises the step of:

3   storing the copy as one or more large binary objects in the second database.

1   55.   (New) The method of claim 52, wherein the step of storing the copy comprises the step

2   of:

3   storing the copy in the first database.

1   56.   (New) The method of claim 55, wherein step of storing the copy in the first database

2   comprises the step of:

3   storing the copy as one or more large binary objects in the first database.

1   57.   (New) The method of claim 42, wherein the copy is a first copy and the set of two or

2   more processes is a first set of two or more processes, and wherein the method further

3   comprises the steps of:

4   causing the first copy to be stored into one or more files by the first set of processes; and

5   requesting each process of a second set of one or more processes to load a second copy

6   into a second database, wherein the second copy is based on the first copy stored

7   in the one or more files.

1   58.   (New) The method of claim 57, wherein:

2   the one or more files include one or more flat files;

3   the first set of two or more processes includes a set of two or more snapshot worker

4   processes;

5   the second set of one or more processes includes a set of one or more loader worker

6   processes; and

7   the second database is a snapshot table database.

1   59.   (New) The method of claim 57, further comprising the steps of:

2   each process of the first set of processes informing a coordinator process when said each

3   process of the first set of processes has finished writing to a particular file of the

4   one or more files; and

5   after each process of the first set of processes informs the coordinator process, the

6   coordinator process informing each process of the second set of processes when

7   said each process of the second set of processes can begin loading the second

8   copy to the second database.

1   60.   (New) The method of claim 57, further comprising the steps of:

2   allowing an application to read the second copy that is stored in the second database,

3   wherein the application generates a schedule based on data within the second

4   copy.

1   61.   (New) The method of claim 42, further comprising the steps of:

2   causing the first copy to be stored into one or more files by the set of processes; and

3   allowing an application to read the first copy that is stored in the one or more files,

4   wherein the application generates a schedule based on data within the first copy.

1 62.   (New) The method of claim 42, further comprising the step of:

2   using a list of database tables to identify the portion of the first database.

1 63.   (New) The method of claim 62, wherein the set of processes uses the list of database

2   tables to extract the copy from the first database.

1 64.   (New) The method of claim 62, wherein the list of database tables is a copy table list that

2   is generated by one process of the set of processes, and wherein the method further

3   comprises the steps of:

4   requesting each process of a third set of processes to identify and delete, based on the

5   copy table list, data from a second database.

1 65.   (New) The method of claim 62, further comprising the steps of:

2   communicating with an application to identify a set of data for generating a schedule; and

3   causing the list of database tables to be created based on the identified set of data.

1 66.   (New) A computer-readable medium carrying one or more sequences of one or more

2   instructions for extracting data from a first database, the one or more sequences of one or

3   more instructions including instructions which, when executed by one or more

4   processors, cause the one or more processors to perform the steps of:

5   causing a set of two or more processes to obtain snapshot times that cause all subsequent

6   reads of the first database by each process in the set of processes to return data

7   from a same state of at least a portion of the first database as will be returned to

8   all other processes in the set of processes; and

9   causing the set of processes to extract a copy of the portion of the first database.

1   67.   (New) The computer-readable medium of claim 66, wherein the set of processes are

2   included in a plurality of processes, and the step of causing the set of processes to obtain

3   snapshot times comprises the step of:

4   causing the plurality of processes to attempt to obtain snapshot times that cause all

5   subsequent reads of the first database by each process of the plurality of processes

6   that successfully obtains one of the snapshot times to return data from the same

7   state of at least the portion of the first database as will be returned to all other

8   processes of the plurality of processes that successfully obtain one of the snapshot

9   times.

1   68.   (New) The computer-readable medium of claim 66, wherein at least one process of the

2   plurality of processes fails to one of the snapshot times within a particular period of time.

1   69.   (New) The computer-readable medium of claim 66, wherein each process of the set of

2   processes obtains a snapshot time by sending a request for a snapshot time to a database

3   management system that is associated with the first database and that assigns a snapshot

4   time in response to the request.

1   70.   (New) The computer-readable medium of claim 66, wherein each process of the set of

2   processes is assigned to extract a particular subset of the portion from the first database.

1   71.   (New) The computer-readable medium of claim 66, wherein each process of the set of

2   processes obtains a snapshot time within a particular period of time.

1   72.   (New) The computer-readable medium of claim 71, wherein the set of processes does not

2   begin to extract the copy of the portion until after expiration of the particular period of

3   time.

1   73.   (New) The computer-readable medium of claim 71, wherein at least one process of the

2   set of processes begins to extract the copy of the portion prior to expiration of the

3   particular period of time.

1   74.   (New) The computer-readable medium of claim 66, further comprising instructions for

2   performing the step of:

3   locking the portion of the first database during the particular period of time.

1   75.   (New) The computer-readable medium of claim 66, further comprising instructions for

2   performing the step of:

3   using a set transaction read only command such that each process of the set of processes

4   can obtain a snapshot time.

1   76.   (New) The computer-readable medium of claim 66, further comprising instructions for

2   performing the step of:

3   storing the copy separate from the portion.

1   77.   (New) The computer-readable medium of claim 76, wherein the step of storing the copy

2   comprises the step of:

3   storing the copy in a second database.

1   78.   (New) The computer-readable medium of claim 77, wherein step of storing the copy in

2   the second database comprises the step of:

3   storing the copy as one or more large binary objects in the second database.

1   79.   (New) The computer-readable medium of claim 76, wherein the step of storing the copy

2   comprises the step of:

3   storing the copy in the first database.

1   80.   (New) The computer-readable medium of claim 79, wherein step of storing the copy in

2   the first database comprises the step of:

3   storing the copy as one or more large binary objects in the first database.

1   81.   (New) The computer-readable medium of claim 66, wherein the copy is a first copy and

2   the set of two or more processes is a first set of two or more processes, and further

3   comprising instructions for performing the steps of:

4   causing the first copy to be stored into one or more files by the first set of processes; and

5   requesting each process of a second set of one or more processes to load a second copy

6   into a second database, wherein the second copy is based on the first copy stored

7   in the one or more files.

1   82.   (New) The computer-readable medium of claim 81, wherein:

2   the one or more files include one or more flat files;

3   the first set of two or more processes includes a set of two or more snapshot worker

4   processes;

5   the second set of one or more processes includes a set of one or more loader worker

6   processes; and

7   the second database is a snapshot table database.

1   83.   (New) The computer-readable medium of claim 81, further comprising instructions for

2   performing the steps of:

3   each process of the first set of processes informing a coordinator process when said each

4   process of the first set of processes has finished writing to a particular file of the

5   one or more files; and

6   after each process of the first set of processes informs the coordinator process, the

7   coordinator process informing each process of the second set of processes when

8   said each process of the second set of processes can begin loading the second

9   copy to the second database.

1   84.   (New) The computer-readable medium of claim 81, further comprising instructions for

2   performing the step of:

3   allowing an application to read the second copy that is stored in the second database,

4   wherein the application generates a schedule based on data within the second

5   copy.

1   85.   (New) The computer-readable medium of claim 66, further comprising instructions for

2   performing the steps of:

3   causing the first copy to be stored into one or more files by the set of processes; and

4   allowing an application to read the first copy that is stored in the one or more files,

5   wherein the application generates a schedule based on data within the first copy.

1   86.   (New) The computer-readable medium of claim 66, further comprising instructions for
2   performing the step of:
3   using a list of database tables to identify the portion of the first database.

1   87.   (New) The computer-readable medium of claim 86, wherein the set of processes uses the
2   list of database tables to extract the copy from the first database.

1   88.   (New) The computer-readable medium of claim 86, wherein the list of database tables is
2   a copy table list that is generated by one process of the set of processes, and further
3   comprising instructions for performing the step of:
4   requesting each process of a third set of processes to identify and delete, based on the
5   copy table list, data from a second database.

1   89.   (New) The computer-readable medium of claim 86, further comprising instructions for
2   performing the steps of:
3   communicating with an application to identify a set of data for generating a schedule; and
4   causing the list of database tables to be created based on the identified set of data.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

To the extent necessary, a petition for an extension of time under 37 C.F.R. § 1.136 is hereby made. Please charge any fee shortages or credit any overages Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Craig G. Holmes
Reg. No. 44,770

1600 Willow Street
San Jose, CA 95125
(408) 414-1080, ext. 207
Date: December 13, 2001
Facsimile: (408) 414-1076
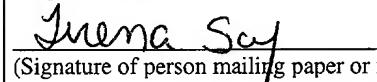
EXPRESS MAIL CERTIFICATE OF MAILING
"Express mail" mailing label number <u>EL734970286US</u>
Date of deposit <u>December 13, 2001</u>
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner for Patents, BOX REISSUE, Washington, D.C. 20231.
<u>Tirena Say</u>
(Typed or printed name of person mailing paper or fee)
Tirena Say
(Signature of person mailing paper or fee)

Application of Nadeem SYED, et al., Ser. No. NYA
Separate Paper with Statement of Status and Explanation of Support for All Changes to the Claims
to Accompany Reissue Patent Application

STATEMENT OF STATUS AND EXPLANATION OF SUPPORT FOR ALL

CHANGES TO THE CLAIMS AS REQUIRED BY 37 C.F.R. §1.173(c)

TO ACCOMPANY REISSUE PATENT APPLICATION

STATUS OF CLAIMS

Claims 1 - 89 are **pending** in the application after the entry of this amendment.
No claims have been **cancelled**.

Claims 7, 12, 15, 26, 28, 29, 36, 38, and 39 have **NOT been amended**.

Claims 9, 11, 13, 14, 16-21, 27, 30, 31, 37, 40, and 41 have been **amended** to make *minor editorial changes* and are fully supported by at least their respective original claims in the disclosure.

Claims 1-6, 8, 10, 22-25, and 32-35 have been **amended** by this preliminary amendment and are supported by at least the portions of the disclosure listed below.

Claims 42-89 have been **added** by this preliminary amendment and are supported by at least the portions of the disclosure listed below.

SUPPORT FOR CHANGES TO CLAIMS

As noted above, Claims 9, 11, 13, 14, 16-21, 27, 30, 31, 37, 40, and 41 have been amended to make minor editorial changes and are fully supported by at least their respective original claims in the disclosure.

Claims 1, 22, 32 are supported by at least FIGS. 2, 3A, 3B, 4, 5A, and 5B and the related discussion in the specification in Columns 5-13. For example, those portions of the disclosure describe embodiments that include either a snapshot worker 206 (FIGS. 4, 5A, and 5B) or a set of snapshot workers 206-214 (FIGS. 2, 3A, and 3B) that obtain snapshot times (e.g., steps 312, 314 and 510, 512). The snapshot worker(s) then retrieve the data from the database 204 (e.g., steps 318 and 518), store the data in a set of flat files 220-226 (e.g., 322 and 518), and then a set of loader workers 230-236 store a copy of the data in the snapshot table database 240 (e.g., steps 328 and 524). The memory based planner 202 then accesses the copy of the data from snapshot table database 240 (e.g., steps 336 and 532).

Application of Nadeem SYED, et al., Ser. No. NYA
Separate Paper with Statement of Status and Explanation of Support for All Changes to the Claims
to Accompany Reissue Patent Application

Claims 2, 23, 33 are supported by at least FIGS. 2 and 4 that depict snapshot table database 240 storing a copy of the data extracted from database 204, such as by following the examples of FIGS 3A and 3B and FIGS 5A and 5B, respectively.

Claims 3, 24, 34 are supported by at least FIGS. 2 and 4 that depict a set of flat files 220-226 and loader workers 230-236, and by FIGS. 3A and 3B and FIGS. 5A and 5B, respectively, that depict loader workers 230-236 copying the data from flat files 220-228 to snapshot table database 240 (e.g., steps 328 and 524).

Claims 4, 25, 35 are supported by at least FIGS. 3A, 3B, 5A, and 5B that described snapshot worker 206 or snapshot workers 206-214, respectively, informing coordinator 216 that the data from database 204 has been copied to flat files 220-226 (e.g., steps 322 and 518). Further, FIGS. 3A, 3B, 5A, and 5B describe coordinator 216 directing loader workers 230-236 to copy data from flat files 220-226 to snapshot table database 240 (e.g., steps 328 and 524) after coordinator 216 is informed that snapshot workers 206-214 have copied the data to flat files 220-226 (e.g., steps 322 and 518).

Claim 5 is supported by at least FIGS. 2 and 4 that depict a plurality of flat files 220-226 and a plurality of loader processes 230-236.

Claim 6 is supported by at least FIGS. 2, 3A, 3B, 4, 5A, and 5B that describe copying data from database 204 (e.g., steps 322 and 518), which creates a first copy data in flat files 220-226, and then copying the data from flat files 220-226 (e.g., steps 328 and 524), which creates a second copy of the data in snapshot table database 240.

Claim 8 is supported by at least FIGS. 2 and 4 that depict a set of delete workers 242-248.

Claim 10 is supported by at least FIGS. 2, 3A, 3B, 4, 5A, and 5B that describe a set of flat files 220-226 and the memory based planner 202 being notified that the data has been copied to snapshot table database 240 (e.g., steps 334 and 530), with memory based planner 202 then generating a planning schedule based on the data in snapshot table database 240 (e.g., steps 336 and 532).

Claims 42 and 66 are supported by at least FIGS. 2, 3A, and 6 that describe coordinator 216 requesting a plurality of snapshot workers 206-215 to obtain snapshot times (e.g., step 312) and then snapshot workers 206-214 retrieving a copy of the data from database 204 (e.g., step 318). Consistency of the state of the database between the

Application of Nadeem SYED, et al., Ser. No. NYA
Separate Paper with Statement of Status and Explanation of Support for All Changes to the Claims
to Accompany Reissue Patent Application

set of snapshot times may be ensured in a number of ways, including but not limited to the locking of the database for a time period during which snapshot times are obtained or by using a "set transaction read only" command, as described in Columns 6 and 7.

Claims 43 and 67 are supported by at least FIGS. 2A and 2B and the related discussion in Columns 6 and 7 of the specification that explain that the snapshot times are successfully obtained by snapshot workers 206-214, but not snapshot worker 215. As a result, snapshot workers 206-214 will obtain data from the same state of database 204.

Claims 44 and 68 are supported by at least Column 6, line 51 to Column 7, lines 8 describes the use of a "set transaction read only" command to ensure that all subsequent reads by a process for a transaction return data as of the time of the command.

Claims 45 and 69 are supported by at least Column 5, line 66 - Column 6, line 8 and Column 8, line 65 - Column 9, line 10 that explain that the snapshot workers obtain snapshot times from the database management system (DBMS) that assigns the snapshot times when the snapshot workers solicit the snapshot times from the DBMS.

Claims 46 and 70 are supported by at least FIG. 3A that describes the coordinator 216 coordinating the retrieval of the data from database 204 by snapshot workers 206-214 (e.g., step 318). As described in the specification in Column 9, lines 14-22, each snapshot worker is assigned a portion of the data to be copied. For example, snapshot worker 206 may be assigned to copy database table "17" from database 204.

Claims 47, 48, 71, and 72 are supported by at least Column 7, lines 9-20, which describes that snapshot workers 206-214 are successful snapshot workers because their snapshot times were obtained prior to the expiration of T_WAIT. In contrast, snapshot worker 215 does not succeed because snapshot worker 215 is assumed to have not obtained a snapshot time prior to the expiration of T_WAIT.

Claims 49, 50, 73, and 74 are supported by at least by FIG. 3A that describes coordinating retrieval of the data from a database (e.g., step 318) after releasing the locks on the tables (e.g., step 316) and by Column 7, lines 28-47 that describes an embodiment in which snapshot workers are not required to wait until the end of the T_WAIT period of time before beginning to retrieve data from database 204.

Application of Nadeem SYED, et al., Ser. No. NYA
Separate Paper with Statement of Status and Explanation of Support for All Changes to the Claims
to Accompany Reissue Patent Application

Claims 51 and 75 are supported by at least FIG. 3A that describes locking the tables identified in the data copy table list (e.g., step 310) and by Column 6, lines 39-50 that describe the snapshot times being assigned after the desired tables are locked and before the locks are released (e.g., T_WAIT).

Claims 52 and 76 are supported by at least FIGS. 2, 3A, 4, and 5A that describe copying data from database 204 to flat files 220-226 (e.g., steps 322 and 518) and by original Claims 11, 27, and 37 that feature "storing the copy of the first set of data separate from said first [set] of data."

Claims 53 and 77 are supported by at least FIG. 6 and Column 14, lines 14-25 that describe snapshot workers 206-214 storing the copy of the data from database 204 directly in snapshot table database 240.

Claims 54 and 78 are supported by at least Column 14, lines 26-34 that describe data from database 204 directly in snapshot table database 240 as "large binary objects" (BLOBS).

Claims 55, 56, 79, and 80 are supported by at least Column 14, lines 35-42 that describe snapshot workers 206-214 retrieving data from database 204 and storing a copy as separate data back into database 204, such as in the form of BLOBS.

Claims 57 and 81 are supported by at least FIGS. 2, 3A, 3B, 4, 5A, and 5B and related discussion in the specification in Columns 5-13. For example, those portions of the disclosure describe embodiments that include either a snapshot worker 206 (FIGS. 4, 5A, and 5B) or a set of snapshot workers 206-214 (FIGS. 2, 3A, and 3B) that obtain snapshot times. The snapshot workers then retrieve the data from the database 204, store the data in a set of flat files 220-226, and then a set of loader workers 230-236 store a copy of the data in the snapshot table database 240. The memory based planner 202 then accesses the copy of the data from snapshot table database 240.

Claims 58 and 82 are supported by at least FIG. 2 that depicts flat files 220-226, snapshot workers 206-214, loader workers 230-238, and snapshot table database 240.

Claims 59 and 83 are supported by at least FIGS. 3A and 3B and FIGS. 5A and 5B that describe snapshot worker 206 or snapshot workers 206-214, respectively, informing coordinator 216 that the data from database 204 has been copied to flat files 220-226 (e.g., steps 322 and 518). Further, FIGS. 3A and 3B and FIGS. 5A and 5B

Application of Nadeem SYED, et al., Ser. No. NYA
Separate Paper with Statement of Status and Explanation of Support for All Changes to the Claims
to Accompany Reissue Patent Application

describe coordinator 216 directing loader workers 230-236 to copy data from flat
files 220-226 to snapshot table database 240 (e.g., steps 328 and 524) after
coordinator 216 is informed that snapshot workers 206-214 have copied the data to flat
files 220-226 (e.g., steps 322 and 518).

Claims 60 and 84 are supported by at least FIGS. 2, 3A, 3B, 4, 5A, and 5B that
describe memory based planner 202 generating a planning schedule based on the data in
snapshot table database 240 (e.g., steps 336 and 532).

Claims 61 and 85 are supported by at least FIGS. 2, 3A, 3B, 4, 5A, 5B, and
Column 10, lines 50-56 that described memory based planner 202 using the information
in flat files 220-226 to generate the planning schedule (e.g., step 336)

Claims 62, 63, 86, and 87 are supported by at least Column 5, lines 41-65 and
Column 6, lines 21-46 that describe how generation of a planning schedule typically
requires data from a subset of the tables in database 204, the use of a copy table list 218
that lists the database tables (1…N) in database 204 that are needed for the planning
schedule (e.g., the "desired tables"), the locking of the identified tables, and the retrieval
of data from the desired tables by the snapshot workers.

Claims 64 and 88 are supported by at least FIGS. 2, 3A, 3B, 4, 5A, 5B and
Column 8, lines that describe launching delete workers 242-248 using copy table list 218
to delete stale data in the snapshot tables in snapshot table database 240 (e.g., steps 320
and 516).

Claims 65 and 89 are supported by at least FIGS. 3A and 5A that describe
snapshot worker 206 determining which tables of database 204 are required by the
memory based planner 202 for generation the planning schedule and generating copy
table list 218 based on this information (e.g., steps 306 and 506), and by original
Claims 9 that features "communicating with the software application to identify a set of
planning data…and creating the copy table list based on the identified set of planning
data."